

SURVEY RESULTS show that the top FIVE items of interest rank as follows:

Plug In Language Cartridge
Data Processing Capability
Computer Control of Tape Machine
Modem

tie { Self-Diagnostic Cartridge
Word Processing Capability

Hardly mentioned were Bally/Bally, plotter, and vocal communications Readers are obviously interested in being able to do things with the Bally - to make it into a real-life computer, so we will investigate the above list.

FIORYTHM CORRECTION noted on p. twenty-six resulted from a subscriber question, which has the underlying question of 'how do I solve programming glitches?' and I thought I'd describe the process used in this specific example (as opposed to looking up the original listing, which you can't do).

The first area to note is that the command involves a subroutine. What subroutine? Well, solve the equation given, $Mx10+20$. First look for M. In line 40, you see that M represents the Month of birth, and so it has a value of 1 through 12. Solving the equation yields twelve numbers, from 30 to 140, which represent line numbers. Looking at lines 30 through 140 does not show much of value. Here is where intuition, or experience, or whatever, helps. The purpose of the program is to develop data based on the number of days between the date of birth and some time in the present...so you need a program that adds up the following: the days in the year of birth from the day of birth to the end of the year; the days in all the years from the year of birth to this year; and finally the number of days from Jan 1 to the required day. Since our questioned equation uses M, it is involved in the calculation to determine the number of days from the month of birth to the end of the birth year. This could be done by multiplying the number of months left by 30 and doing something about 28- and 31-day months, or you could set up a little table having the number of days from each month to the end of the year and have the program use a subroutine to get to the right spot in the table. And there are two tables like that in the program. One is from lines 260 to 370, and the other from 510 to 610. By inspection, we can see that if line 101 had an error, and we wanted to go to line 260 for month 1, the line would have to be $1 \times 10 + 250$. And similarly for month 12-we see that there was a mis-type. Double check line 510 and you'll see the equation would have to be $1 \times 10 + 490$, hard to mis-type that one.

POKE has been extensively researched by Dave Clark in the area $\$(20050)=n$ to $\$(20076)=n$. Two useful POKE's are $\$(20070)$ and $\$(20076)$ which replace standard keyboard characters with new geometric shapes - these could be used as game pieces. Try

```
10 FOR A = 0 TO 255
20  $\$(20070) = 160$ ; PRINT #1,A,
30  $\$(20070) = A$ ; PRINT #1,1,
40 NEXT A
```

You will see a

run of weird characters, the our familiar ones, then more unusual ones. The POKE at (20076) is similar, Substitute

```
20  $\$(20076) = 8$ ; PRINT #1,A,
30  $\$(20076) = A$ ; PRINT #1,1,
```

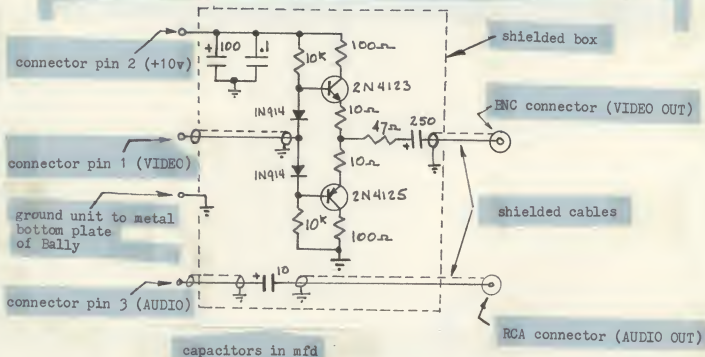
Dave also sent a program to draw a circle , as follows

```
10 INPUT X,Y          40 A = X + Y ÷ 2; E = Y - A ÷ 2
20 CLEAR              50 IF PX(A,B) = 1 GOTO 10
30 BOX X,Y,1,1,1      60 X = A; Y = B; GOTO 30
```

Change the divisors in line 40 to get oval-shaped figures.

DIRECT VIDEO CIRCUIT DIAGRAM (monochrome) is included below. Designed and built by Dan Sandin, it was submitted by Phil Morton. I built one and tried it out on my b/w monitor. (There is no color capability in this circuit). My breadboard layout works, but needs proper shielding for best results. Items to keep in mind:

- Use shielded coax - RG 174/U recommended.
- Remove Bally plastic upper cover and pull off (to the side) the RF modulator (tin box, 2x3-3/4") on left. Pin numbers are 1 to 8, with 1 towards the front.
- ENC connector is standard for video. Audio could be either ENC or RCA
- Don't ask me how to hook it up to your TV.
- If there is enough interest, we could make up an inexpensive kit (say \$15 tops)



transistor substitutes:

2N4123 = RCA SK3444 = Motorola HEP S0015
 2N4125 = RCA SK3466 = Motorola HEP S0029

CONGRATULATIONS are in order to the 12 newly created Arcadians in the Digital Video course at the School of Art Institute of Chicago. They used six Arcades (w/Basic) to learn programming.

SPACE WAR by Dave Clark. Listing on p.thirty one. The object is to destroy the nasty Klingons again. They must be friends of the Empire, because they too have a Death Star, located in one of the quadrants. Shooting it with a torpedo will get you recharged and reloaded with torpedos. K=Klingon, \$ = Starbase, S=Klingon Starbase, * = star. Command 1 raises/lowers deflector shields. Command 2 fires phaser. Command 3 fires torpedo. Command 4 moves you into one of the nine (3x3) quadrants, first enter x-value (1 to 3) then y-value (1 to 3). Command 5 allows you to self-destruct.

arcadian

EDITORIAL As people write to me about one thing or another, I am finding that there is a substantial number to whom computing is a mystery. Perhaps they picked up an Arcade for its game capability, and then discovered the TASIC cartridge - or they saw this as an inexpensive way to get into the computer world - or whatever. I am just a neophyte myself, but we shall try a few tutorials to outline the basics. I trust that the professionals amongst us will look the other way as I plod ahead.

MEMORY TUTORIAL The key factor in any computer is its memory banks. In general, the more memory the machine has, the better it is because it indicates the amount of work that can be done. Memory in general can be considered as a gigantic bunch of boxes, something on the order of the post office system of sorting mail. Each box has an address in accordance with the following. All memory boxes are set up in a long rectangular shape, 16 boxes across, and down as far as necessary. Numbering of the boxes (that is, the address) starts at the upper left corner and goes over to the right in the top row to the 16th box, then drops a row and starts again at the left, and on and on. The first couple of rows would look like this:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F
2	0010	0011	0012	0013	0014	0015	0016	0017	0018	0019	001A	001B	001C	001D	001E	001F
3	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F

Note that we start using letters in the last six boxes. This notation is called hexadecimal. Now, as we get down to row 11, the hexadecimal notation takes over again, and follows on to the end. At row 11 we have:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
10	0090	0091	0092	0093	0094	0095	0096	0097	0098	0099	009A	009B	009C	009D	009E	009F
11	00A0	00A1	00A2	00A3	00A4	00A5	00A6	00A7	00A8	00A9	00AA	00AB	00AC	00AD	00AE	00AF
12	00B0	00B1	00B2	00B3	00B4	00B5	00B6	00B7	00B8	00B9	00BA	00BB	00BC	00BD	00BE	00BF

And as we get to the bottom of the array, we find that the biggest number you can get is FFFF (hex), or $(16 \times 16 \times 16 \times 16) = 65,536$ in normal numbers. This is commonly called 64000, or 64K in computerese. Each box can contain one piece of data called a BYTE, so the above describes a memory of 64K bytes.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	FFFF	FFF1	FFF2	FFF3	FFF4	FFF5	FFF6	FFF7	FFF8	FFF9	FFFA	FFFB	FFFC	FFFD	FFFE	FFFF

last box →

A sub-specification of memory is the type, which divides into two categories: that which is fixed; and that which can be changed.

FIXED - this kind of memory cannot be changed once it is implanted - nor added to, or subtracted from, etc. It can only be looked at, or read. So they call it **READ ONLY MEMORY**, or **ROM**. If there is an Operating System included in a computer, it is in ROM because this allows the computer to be ready to operate as soon as you turn it ON. In the Bally, there is an

8K ROM chip that contains the resident games, controller and keypad operations, etc. The addresses of the bytes in there are from 0000 to 1FFF in hex notation (therefore $2 \times 16 \times 16 \times 16 = 8192$). As mentioned above, you can read the bytes that are permanently stored in the ROM addresses - here is line 0220 (addresses from 0220 through 022F):

0220	0221	0222	0223	0224	0225	0226	0227	0228	0229	022A	022B	022C	022D	022E	022F
5B	2F	53	43	4F	62	45	4B	23	2F	4F	46	2F	5B	4C	41

exciting, isn't it? I have it on good authority that this translates to:

X S C O R E # O F P L A

Programs to read memory have been printed in Volume I.

CHANGEABLE - In order to do useful work, there must be a memory area that can contain the program you want to operate, plus the data that you are going to utilize, input, or store. The computer provides a block of empty boxes in a space called **RANDOM ACCESS MEMORY**, or **RAM**. In some computers, the Operating System is stored on a tape or disc, and when you turn on the machine, you have to transfer the Operating System from storage to a portion of the RAM that is set aside for this purpose.

In the Bally, when you insert the **BALLY BASIC** cartridge, 4K of ROM is added which contains the **Tiny BASIC** Operating System, plus 1.8 K of RAM for programming. In this instance, the amount of RAM is totally out of proportion to the amount of ROM, and that is what we are attempting to rectify with **PROJECT ONE**.

XY TUTORIAL is a software program with documentation, sold by **SEBREE's**. We have received a copy for review, and I've finally decided that the review form we have is not adequate for this kind of material.

This program teaches the user some of the mysteries of the **XY** command and shows him how to use it. The 12 pages of documentation takes one through, in step by step fashion, a number of programs (5) that are essentially put together in program 6, into a 3-d display (of which one picture was shown under their ad on p. twenty-six). One of the documentation pages gives a map of the screen necessary to utilize the **XY** properly.

The program is excellent for those who want to develop graphical displays or games; those interested in game playing would find the display somewhat interesting but not directly useful. Again, it is a tutorial, not a game.

MICHIGAN user group notes that they will have another meeting on April 12, at noon, at the George Moses Co. 110 East North St., Brighton, MI. The telephone is 313-227-1575.

CROSS COUNTRY PROGRAM TRANSMISSION was accomplished on Jan 12 when Larry Camnitz sent the program "Connect Four" to me via telephone. Larry used a direct tie to the 'phone line from his recorder output, through a capacitor; on my end I used a Radio Shack gadget #43-236 which picks off a signal to enter the recorder directly. I also have a circuit here for a similar scheme using opto-isolators, that just arrived. I plan to look it over for the next issue.



arcadian

```

1
10 F=RND (3);G=F;I=RND (3);J=RND (3);K=1;M=RND (16)+16;&(23)=255
20 FC=7;BC=8;P=0;C=1200;D=8;E=0;O=0;CLEAR
25 V=RND (3);W=RND (3);Q=RND (4);IF M<3W=M
30 GOSUB 860;GOSUB 880;CY=40;PRINT " 1 2 3 4 5 6 7";FOR H=1TO 7;PRINT #1,H;NEXT
T H
40 CY=24;CX=10;PRINT "QDRNT ",#1,F,"-",G
50 CX=10;PRINT "ENRGY ",#1,C;CX=10;PRINT "TORPS ",#1,D;CX=10;PRINT "SHLDS ",#1
,E;CX=10;PRINT "KLGOS ",#1,M,"-",W
53 IF 0 IF KGOSUB 880;CY=8;CX=-65;PRINT "DEATH STAR";GOTO 140
56 IF E<0PRINT "ENTERPRISE DESTROYED!";GOTO 1200
60 IF Q=0GOTO 75
70 FOR H=1TO Q;GOSUB 120;PRINT "*",;NEXT H
75 IF W=0GOTO 90
80 FOR H=1TO W;GOSUB 120;PRINT "K",;NEXT H
90 IF V=3GOSUB 120;PRINT "$",
95 IF P GOTO 110
100 IF W=1GOSUB 120;PRINT "S",
110 GOTO 140
120 R=RND (7);b12=83;X=RND (7);b8=24;CX=R;CY=X;RETURN
140 N=-36;T=8;GOSUB 890
150 CX=-77;CY=-24;INPUT "COMMAND:"T;IF (T<1)+(T>5)GOTO 150
153 IF (T=1)+(T=5)GOTO 160
156 IF C<1GOTO 150
160 GOTO Tb200
200 INPUT "UNITS TO SHIELDS"H
210 IF (H<0)+(C+E-H<0)GOTO 130
220 C=C+E-H;E=H;GOTO 30
400 IF W#0GOTO 410
405 PRINT "WEAPON:NEGATIVE";GOTO 30
410 INPUT "UNITS TO FIRE"U
420 IF (U<1)+(C-U<0)GOTO 30
430 C=C-U;T=W;FOR H=1TO W;GOSUB 860;IF U<W-QbRND (5)>100GOTO 450
440 PRINT "NOT ENOUGH FIRE-POWER";NEXT H;GOTO 460
450 PRINT "KLINGON DESTROYED";M=M-1;W=W-1;NEXT H
460 E=E-U<T-QbRND (5);bT
470 GOSUB 850;GOTO 30
500 IF (W=0)+(D=0)GOTO 405
610 INPUT "TRAJ. X"R;CY=-32;INPUT "TRAJ. Y"X;IF (R<1)+(R>8)+(X<1)+(X>8)GOTO 30
620 N=-30;T=8;GOSUB 890;N=Rb12=84;T=(8-X)b8=24;GOSUB 890;CY=-32;PRINT "TYPE 1 T
O FIRE
640 IF KP&(21)=255;GOSUB 860;IF 0 IF K PRINT "DEATH STAR HIT!";C=-1;K=0;GOTO 69
0
650 IF PX(N-2,T-3)=1M=M-1;W=W-1;GOSUB 870
660 IF PX(N-1,T-3)=1P=1;M=M-1;GOSUB 870
670 IF PX(N-1,T-2)=1V=2;PRINT "WHO'S SIDE ARE YOU ON?"
680 IF PX(N-1,T-1)=1 Q=Q-1;GOSUB 870
690 D=D-1;FOR H=1TO 100;BOX N,T,5,7,3;NEXT H;&(21)=0
700 E=E-(100bW+RND (50));cZ
710 IF C=-160GOTO 20
720 GOSUB 850;GOTO 30
800 INPUT "QUADRANT X"Y
810 GOSUB 860;INPUT "QUADRANT Y"U;GOSUB 860
820 IF (Y<1)+(Y>3)+(U<1)+(U>3) GOTO 30
825 IF F=Y IF G=U PRINT "ILLEGAL MOVE";GOTO 30
830 O=0;IF K IF Y=I IF U=J FOR H=1TO 10;GOSUB 860;PRINT "T00000000M";NEXT H;O=
1
840 C=C-ABS(F-Y)b10-ABS(G-U)b10;F=Y;G=U;GOTO 25
850 IF M=0PRINT "WELL DONE,ALL KLINGONS DESTROYED!";GOTO 1200

```

Space War by Dave Clark
text on p. twenty eight

```

855 RETURN
860 CY=-32;BOX 0,-32,160,24,2;RETURN
870 PRINT "DIRECT HIT!";RETURN
880 BOX -35,8,78,56,2;RETURN
890 BOX N,T,5,1,2;BOX N,T,1,5,2;RETURN
1000 CLEAR;PRINT "YOU HAVE INSTRUCTED THE
ON-BOARD COMPUTER TO SELF DESTRUCT!
1200 PRINT "SPACE WAR IS OVER!"

```



```

2 .
3 :RETURN
5 .
10 CLEAR :GOSUB 900
20 GOTO 80
30 CY=16;PRINT F," BUSHELS OF FOOD";RETURN
40 CY=8;PRINT A," ACRES OF LAND";RETURN
50 PRINT "HOW?";RETURN
60 FOR E=1TO 2000:NEXT E;RETURN
70 CY=-24;PRINT " ";RETURN
80 P=100;F=3000;A=1000;Y=0;L=0;NT=0;W=0;K=0;N=18+RND (3)
90 CLEAR
100 PRINT "KING HAMURABI
110 IF Y>0GOTO 140
120 PRINT "YOU START YOUR RULE WITH:
130 GOTO 180
140 IF Y=10GOTO 170
150 PRINT "AFTER ",#1,Y," YEAR(S) OF RULE:
160 GOTO 180
170 PRINT "AT THE END OF YOUR REIGN:
180 PRINT P," PEOPLE
190 GOSUB 30;GOSUB 40
200 IF Y#10GOTO 280
210 PRINT K," PEOPLE DIED
220 PRINT
230 Z=(Pb2)+(F-(Pb20))c10)+(Ab10)cP)-((K-10)b5)
240 PRINT Z," PERFORMANCE PTS.
250 IF Z>M M=Z
260 PRINT "TODAY'S HIGH SCORE=",#5,M
270 PRINT "AGAIN?";A=KP;GOTO 10
280 V=17+RND (5)
290 PRINT V," BHL/AC LAND VALUE
300 PRINT "BUY LAND HOW MANY ACRES?
310 INPUT B
320 IF B=0GOTO 410
330 L=BbV
340 IF L<FGOTO 380
350 IF L>FGOSUB 50
360 GOSUB 60;GOSUB 70
370 GOTO 310
380 F=F-L;A=A+B
390 GOSUB 30;GOSUB 40
400 IF B>0GOTO 510
410 CY=-8;PRINT "SELL LAND HOW MANY ACRES?
420 INPUT S
430 IF S=0GOTO 510
440 IF S<0GOTO 490
450 IF S=APRINT "SO YOUR QUITTING!!";GOTO 260
460 IF S>0GOSUB 50
470 GOSUB 60;GOSUB 70
480 GOTO 420
490 A=A-S;F=F+UbS
500 GOSUB 30;GOSUB 40
510 CY=-8;PRINT "GIVE OUT FOOD # BUSHELS? "
520 INPUT C
530 IF C<FGOTO 560
540 IF C>FGOSUB 50
550 GOSUB 60;GOSUB 70;GOTO 520
560 F=F-C;GOSUB 30

```

HAMURABI by Dick Hauser

You are the King, and must make decisions on running the kingdom economically. Start with 100 people, 1000 acres/land & 3000 bushels/food. Buy & sell land, feed people, and plant the crops using food as barter. And things happen. The object is to garner performance points based on how you govern and to figure out how the game works. It takes bushels to plant land and people can plant only so many acres.

If you last 10 years your reign will be complete.

PROGRAM CONTINUED
ON P. 33

```

570 CY=-8;PRINT "PLANT CROPS? # OF ACRES? "
580 GOTO 600
590 GOSUB 60;GOSUB 70
600 INPUT G;X=G*2
610 IF (G>A)+(X>F)+(G>15bP)GOSUB 50;GOTO 590
620 F=F-X;GOSUB 30;GOSUB 60
630 D=P-C*G
640 IF D>0GOTO 660
650 IF D<0D=0;U=RND (3)-1
660 H=RND (4)+1;T=H*G
670 R=((F+T)*100)*RND (7)
680 F=F-R+T;J=RND (9)
690 I=((F*250)*c(D+1))+((AcP)-10)
700 IF I<=50GOTO 720
710 I=50
720 P=P+I-D+U;Y=Y+1;W=W+I;K=K+D;CLEAR
730 PRINT "KING HAMURABI
740 PRINT "LAST YEAR
750 IF D>Pc4IF J#5IF Y>3PRINT D," PEOPLE STARVED      YOU HAVE BEEN THROWN IN
PRISON!";GOTO 260
760 IF D>0PRINT D," DEATHS
770 IF U>0PRINT U," BIRTHS
780 PRINT I," PEOPLE IMMIGRATED
790 IF W>Pc2+Wc3PRINT "IMMIGRANTS EXCEED COUNTRY-MEN AND HAVE TAKEN OVER      THE
COUNTRY!";GOTO 260
800 IF J#5GOTO 850
810 PRINT "THE PLAGUE KILLED
820 Q=(Pb(10bRND (5))*c100
830 PRINT Q," PEOPLE
840 P=P-Q
850 PRINT "YOUR PEOPLE HARVESTED
860 PRINT T," BUSHELS AT
870 PRINT H," BUSHELS PER ACRE
880 PRINT "RATS ATE ",#3,R," BUSHELS
890 GOSUB 60;GOTO 90
900 FOR A=-63TO 57STEP 24;BOX A,32,12,8,1;NEXT A;BOX -3,-4,132,64,1
910 CX=-47;CY=0;PRINT " KING HAMURABI";CX=-47;PRINT " BY R M HOUSER";A=KP;RETUR

```

N

BINGO

```

30 CLEAR ;NT=1;FC=179;K=237-255
40 &(9)=84;&(0)=7;&(1)=7;&(2)=88;&(3)=88
50 BOX -40,9,68,65,1;BOX -40,4,64,49,2
60 BOX 34,9,68,65,1;BOX 34,4,64,49,2
70 CX=-64;CY=36;PRINT "B I N G O";CX=10;PRINT "B I N G O",
80 FOR L=-60TO -21STEP 13;BOX L,10,1,64,1;BOX L+74,1,1,64,1;NEXT L
90 FOR L=-11TO 19STEP 10;BOX -40,L,68,1,1;BOX 34,L,68,1,1;NEXT L
100 C=0;D=0;X=-69;Y=34
110 FOR L=1TO 125:@(L)=0;NEXT L
120 FOR L=75TO 95STEP 5
130 FOR M=1TO 5
140 N=RND (15)+C
150 IF @(L+1)=N)+@(L+2)=N)+@(L+3)=N)+@(L+4)=N)+@(L+5)=N)GOTO 140
160 @(L+M)=N
170 CX=X;CY=Y+(-Mb10);PRINT #2,N,
180 N=RND (15)+C

```

PROGRAM
CONTINUED p.34

arcadian

```

190 IF (@(L+26)=N)+( @(L+27)=N)+( @(L+28)=N)+( @(L+29)=N)+( @(L+30)=N)GOTO 180
195 @(L+25+M)=N
200 CX=X+74;CY=Y+(-Mb10);PRINT #2,N,
210 NEXT M
220 C=C+15;X=X+13
230 NEXT L
232 FOR L=1TO 3;CY=4;CX=-40;PRINT "H",;CX=34;PRINT "C",
233 BOX -40,4,12,9,L;BOX 34,4,12,9,L;NEXT L
234 @(88)=-1;@(113)=-1
240 FOR L=1TO 75
250 N=RND (75);IF @(N)<0GOTO 250
260 @(N)=-1
262 CX=-72;CY=-32
264 IF N<16PRINT "# ",#2,N," UNDER B",
266 IF N>15IF N<31PRINT "# ",#2,N," UNDER I",
268 IF N>30IF N<46PRINT "# ",#2,N," UNDER N",
270 IF N>45IF N<61PRINT "# ",#2,N," UNDER G",
272 IF N>60PRINT "# ",#2,N," UNDER O",
278 BOX 34,-32,70,10,2
280 IF KN(1)<0PRINT " NO? ",
290 IF KN(1)=0PRINT " YES?",
300 IF TR(1)=0GOTO 262
310 IF KN(1)<0GOTO 340
320 A=0;U=0;GOSUB 400
330 GOSUB 600
340 A=25;U=74;GOSUB 400
350 GOSUB 600
360 NEXT L
400 FOR S=76+ATO 100+A;IF @(S)=NGOTO 500
410 NEXT S
430 IF A=25RETURN
440 PRINT " NOPE",;RETURN
500 X=-66
510 FOR S=75+ATO 95+ASTEP 5
512 Y=24
520 FOR T=1TO 5
530 IF @(S+T)=N&(21)=255;BOX X+U,Y,12,9,3;@(S+T)=-1;&(21)=0;RETURN
540 Y=Y-10;NEXT T
550 X=X+13;NEXT S
560 RETURN
600 FOR S=76+ATO 96+ASTEP 5
610 IF @(S)=-1IF @(S+1)=-1IF @(S+2)=-1IF @(S+3)=-1IF @(S+4)=-1GOTO 800
620 NEXT S
630 FOR S=76+ATO 80+A
640 IF @(S)=-1IF @(S+5)=-1IF @(S+10)=-1IF @(S+15)=-1IF @(S+20)=-1GOTO 800
650 NEXT S
660 FOR S=76+ATO 100+ASTEP 6
670 IF @(S)#-1GOTO 700
680 NEXT S
690 GOTO 800
700 FOR S=80+ATO 96+ASTEP 4
710 IF @(S)#-1RETURN
720 NEXT S
800 CY=-40
810 FOR L=0TO 9
820 IF A=0CX=-72;PRINT "BINGO--YOU WIN",
830 IF A=25CX=8;PRINT "BINGO--I WIN",
840 NEXT L

```

BINGO by Ernie Sams

Player card(Human) is green.

Drawn numbers appear on the screen,
use KNob(!) to indicate Yes or No,
pull Trigger to register. Computer
also checks its card.

5. CONNECT FOUR

6. BY L. L. CARNITZ

7. 8 BC=126

8. 20 FC=0

9. 30 NT=0

100 P=1; L=1; M=0; N=0

950 CLEAR; GOSUB 3000

950 CLEAR

1000 BOX -4, -36, 118, 10, 2

1002 CX=-70; CY=20; PRINT "GAME"

1004 CX=45; CY=20; PRINT "BLACK"

1005 CX=58; CY=-20; PRINT #0, M

1006 CX=45; CY=-10; PRINT "WHITE"

1007 CX=58; CY=10; PRINT #0, N

1008 CX=-64; CY=10; PRINT #0, L

1010 BOX -4, -3, 87, 56, 1

1020 FOR B=-40 TO 32 STEP 12

1030 FOR C=20 TO -25 STEP -9

1040 BOX B, C, 11, 8, 2

1050 NEXT C

1060 NEXT B

1070 X=-40

1080 D=-34; E=-34; F=-34; G=-34; H=-34; I=-34; J=-34

1100 CY=40; IF P=1 PRINT "

1110 CY=40; IF P=2 PRINT "

1200 IF TR(1)=1GOTO 2000

1210 IF JX(1)=1GOTO 1500

1220 IF JX(1)=1GOTO 1510

1230 IF JY(2)=1GOTO 3500

1240 GOTO 1200

1400 IF TR(2)=1GOTO 2000

1405 IF &(22)=16GOTO 900

1410 IF JX(2)=1GOTO 1500

1420 IF JX(2)=1GOTO 1510

1430 IF JY(1)=1GOTO 3600

1440 GOTO 1400

1500 X=X+12; GOTO 1600

1510 X=X-12; GOTO 1600

1600 IF X<-40=-40

1610 IF X>32X=32

1620 BOX -4, 30, 100, 6, 2

1700 BOX X, 30, 9, 6, 1

1705 IF P=2BOX X, 30, 5, 4, 2

1706 NT=1

1710 MU="Y"; MU="Z"

1711 NT=0

1720 IF P=1GOTO 1200

1730 IF P=2GOTO 1400

2000 IF X=-400=D+9; GOTO 2200

2010 IF X=-28E=E+9; GOTO 2300

2020 IF X=-16F=F+9; GOTO 2400

2030 IF X=-4G=G+9; GOTO 2500

2040 IF X=8H=H+9; GOTO 2600

2050 IF X=20I=I+9; GOTO 2700

2060 IF X=32J=J+9; GOTO 2800

2200 BOX X, 10, 9, 6, 1

2220 P=2; GOTO 1100

2300 BOX X, E, 9, 6, 1

2310 IF P=2BOX X, E, 5, 4, 2; P=1; GOTO 1100

2320 P=2; GOTO 1100

2400 BOX X, F, 9, 6, 1

2410 IF P=2BOX X, F, 5, 4, 2; P=1; GOTO 1100

2420 P=2; GOTO 1100

2500 BOX X, G, 9, 6, 1

3510 M=N+1; CX=58; CY=-20; PRINT #0, M

3515 L=L+1

3520 IF L=1=KGOTO 4500

3530 IF TR(2)=1GOTO 1000

3540 GOTO 3530

3600 CX=-56; CY=-38; PRINT "BLACK CONNECTS FOUR

3610 N=N+1; CX=58; CY=10; PRINT #0, N

3615 L=L+1

3620 IF L=1=KGOTO 4500

3630 IF TR(1)=1GOTO 1000

3640 GOTO 3630

4500 CY=40; PRINT "GAME OVER"

4510 IF &(22)=16GOTO 900

4520 GOTO 4510

4520 GOTO 1100

3510 M=N+1; CX=58; CY=-20; PRINT #0, M

3515 L=L+1

3520 IF L=1=KGOTO 4500

3530 IF TR(2)=1GOTO 1000

3540 GOTO 3530

3600 CX=-56; CY=-38; PRINT "BLACK CONNECTS FOUR

3610 N=N+1; CX=58; CY=10; PRINT #0, N

3615 L=L+1

3620 IF L=1=KGOTO 4500

3630 IF TR(1)=1GOTO 1000

3640 GOTO 3630

4500 CY=40; PRINT "GAME OVER"

4510 IF &(22)=16GOTO 900

4520 GOTO 4510

4520 GOTO 1100

3510 M=N+1; CX=58; CY=-20; PRINT #0, M

3515 L=L+1

3520 IF L=1=KGOTO 4500

3530 IF TR(2)=1GOTO 1000

3540 GOTO 3530

3600 CX=-56; CY=-38; PRINT "BLACK CONNECTS FOUR

3610 N=N+1; CX=58; CY=10; PRINT #0, N

3615 L=L+1

3620 IF L=1=KGOTO 4500

3630 IF TR(1)=1GOTO 1000

3640 GOTO 3630

4500 CY=40; PRINT "GAME OVER"

4510 IF &(22)=16GOTO 900

4520 GOTO 4510

4520 GOTO 1100

3510 M=N+1; CX=58; CY=-20; PRINT #0, M

3515 L=L+1

3520 IF L=1=KGOTO 4500

3530 IF TR(2)=1GOTO 1000

3540 GOTO 3530

3600 CX=-56; CY=-38; PRINT "BLACK CONNECTS FOUR

3610 N=N+1; CX=58; CY=10; PRINT #0, N

3615 L=L+1

3620 IF L=1=KGOTO 4500

3630 IF TR(1)=1GOTO 1000

3640 GOTO 3630

4500 CY=40; PRINT "GAME OVER"

4510 IF &(22)=16GOTO 900

4520 GOTO 4510

4520 GOTO 1100

3510 M=N+1; CX=58; CY=-20; PRINT #0, M

3515 L=L+1

3520 IF L=1=KGOTO 4500

3530 IF TR(2)=1GOTO 1000

3540 GOTO 3530

3600 CX=-56; CY=-38; PRINT "BLACK CONNECTS FOUR

3610 N=N+1; CX=58; CY=10; PRINT #0, N

3615 L=L+1

3620 IF L=1=KGOTO 4500

3630 IF TR(1)=1GOTO 1000

3640 GOTO 3630

4500 CY=40; PRINT "GAME OVER"

4510 IF &(22)=16GOTO 900

4520 GOTO 4510

4520 GOTO 1100

3510 M=N+1; CX=58; CY=-20; PRINT #0, M

3515 L=L+1

3520 IF L=1=KGOTO 4500

3530 IF TR(2)=1GOTO 1000

3540 GOTO 3530

3600 CX=-56; CY=-38; PRINT "BLACK CONNECTS FOUR

3610 N=N+1; CX=58; CY=10; PRINT #0, N

3615 L=L+1

3620 IF L=1=KGOTO 4500

3630 IF TR(1)=1GOTO 1000

3640 GOTO 3630

4500 CY=40; PRINT "GAME OVER"

4510 IF &(22)=16GOTO 900

4520 GOTO 4510

4520 GOTO 1100

3510 M=N+1; CX=58; CY=-20; PRINT #0, M

3515 L=L+1

3520 IF L=1=KGOTO 4500

3530 IF TR(2)=1GOTO 1000

3540 GOTO 3530

3600 CX=-56; CY=-38; PRINT "BLACK CONNECTS FOUR

3610 N=N+1; CX=58; CY=10; PRINT #0, N

3615 L=L+1

3620 IF L=1=KGOTO 4500

3630 IF TR(1)=1GOTO 1000

3640 GOTO 3630

4500 CY=40; PRINT "GAME OVER"

4510 IF &(22)=16GOTO 900

4520 GOTO 4510

4520 GOTO 1100

3510 M=N+1; CX=58; CY=-20; PRINT #0, M

3515 L=L+1

3520 IF L=1=KGOTO 4500

3530 IF TR(2)=1GOTO 1000

3540 GOTO 3530

3600 CX=-56; CY=-38; PRINT "BLACK CONNECTS FOUR

3610 N=N+1; CX=58; CY=10; PRINT #0, N

3615 L=L+1

3620 IF L=1=KGOTO 4500

3630 IF TR(1)=1GOTO 1000

3640 GOTO 3630

4500 CY=40; PRINT "GAME OVER"

4510 IF &(22)=16GOTO 900

4520 GOTO 4510

4520 GOTO 1100

3510 M=N+1; CX=58; CY=-20; PRINT #0, M

3515 L=L+1

3520 IF L=1=KGOTO 4500

3530 IF TR(2)=1GOTO 1000

3540 GOTO 3530

3600 CX=-56; CY=-38; PRINT "BLACK CONNECTS FOUR

3610 N=N+1; CX=58; CY=10; PRINT #0, N

3615 L=L+1

3620 IF L=1=KGOTO 4500

3630 IF TR(1)=1GOTO 1000

3640 GOTO 3630

4500 CY=40; PRINT "GAME OVER"

4510 IF &(22)=16GOTO 900

4520 GOTO 4510

4520 GOTO 1100

3510 M=N+1; CX=58; CY=-20; PRINT #0, M

3515 L=L+1

3520 IF L=1=KGOTO 4500

3530 IF TR(2)=1GOTO 1000

3540 GOTO 3530

3600 CX=-56; CY=-38; PRINT "BLACK CONNECTS FOUR

3610 N=N+1; CX=58; CY=10; PRINT #0, N

3615 L=L+1

3620 IF L=1=KGOTO 4500

3630 IF TR(1)=1GOTO 1000

3640 GOTO 3630

4500 CY=40; PRINT "GAME OVER"

4510 IF &(22)=16GOTO 900

4520 GOTO 4510

4520 GOTO 1100

3510 M=N+1; CX=58; CY=-20; PRINT #0, M

3515 L=L+1

3520 IF L=1=KGOTO 4500

3530 IF TR(2)=1GOTO 1000

3540 GOTO 3530

3600 CX=-56; CY=-38; PRINT "BLACK CONNECTS FOUR

3610 N=N+1; CX=58; CY=10; PRINT #0, N

3615 L=L+1

3620 IF L=1=KGOTO 4500

3630 IF TR(1)=1GOTO 1000

3640 GOTO 3630

4500 CY=40; PRINT "GAME OVER"

4510 IF &(22)=16GOTO 900

4520 GOTO 4510

4520 GOTO 1100

3510 M=N+1; CX=58; CY=-20; PRINT #0, M

3515 L=L+1

3520 IF L=1=KGOTO 4500

3530 IF TR(2)=1GOTO 1000

3540 GOTO 3530

3600 CX=-56; CY=-38; PRINT "BLACK CONNECTS FOUR

3610 N=N+1; CX=58; CY=10; PRINT #0, N

- For Sale - like new Bally Arcade. Includes 9 tapes (including Bally Basic cartridge), Interface and 4 controllers. Will take highest bid & notify by telephone. Send bid and phone number to
Steve Melton, Garden Grove Apts, 1601-9th St., NW, Gt. Falls, Mont. 59405
- George Collins, 30 Sierra Ave., Piedmont, CA 94611 is expanding his line of software. Send SSAE for more information and a free listing of the program U-BOAT
- Sell Bally computer system complete with Bally Basic and cassette interface; 4 pistol grips and BLACKJACK, POKER, ACEY-DUECY, SEA WOLF, and MISSILE, STARBATTLE. Unit new in Nov '79. Complete 285.
Also brand new hand controls, 10 each, 4 for 30. Bill Fowler, 42 Bent Oak Trail, Fairport, NY 14450

INVASION-REVIEW. This program is based on the "Space Invaders" game found in many arcades. It is a very challenging game, with difficulty levels from one to five. At level five, it is very easy to win, while level one is almost impossible. The major difference between this and the arcade game is speed. This one is considerably slower.

As for criticism, it would be better if the knob control (KN) were given as a better option. A modification is given in the program for its use, but it is not really satisfactory.

Also, perhaps a subroutine could be used to simulate explosions similar to the way the mother ship explodes. However, considering the memory limitations that the Bally has, much further improvement is difficult.

One final comment: Perhaps an option (re-programming) so that the landers would appear not just randomly but in sequence (like the real "Space Invaders") would make for a nicer display and add to the strategy aspect of the game.

Reviewed by Bill Rueger

.....
 Name of Program: Invasion Cassette Name: Invasion
 Description: Game similar to "Arcade Space Invaders"
 Source: George Collins 30 Sierra Ave., Piedmont CA 94611 Price \$5.00
 Reviewed By: Bill Rueger Age: _____
 Ratings : PD= 7 PP= 7 USF= 7 LC= 9 OC= 8 LI= 9 EV= 0 EU= 8 OV= 8 TP= 03 SZ= 34
 Time to Play 2-5 min For Ages ALL # of Players ONE

THIRTY-SIX

ARCADIAN™

Robert Fabris, Tabulator
 3626 Morrie Dr.
 San Jose, CA 95127

FIRST CLASS

MEMORY TUTORIAL

The key factor in any computer is its memory banks. In general, the more memory the machine has, the better it is because it indicates the amount of work that can be done, at least it costs more. Memory in general can be considered as a gigantic bunch of boxes, something on the order of the post office system of sorting mail. Each box has an address in accordance with the following rule. All memory ~~spaces~~^{boxes} are set up in a long rectangular shape, 16 boxes across, and down as far as necessary. Numbering of the boxes (the address) starts at the upper left corner and goes over to the right in the first row to the 16th box, then drops a row and starts again at the left, on and on. The first couple of rows would look like this:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F
0010	0011	0012	0013	0014	0015	0016	0017	0018	0019	001A	001B	001C	001D	001E	001F
0020															003F
↓															↓

Note that we start using letters in the last six boxes. This notation is called hexadecimal

Now as we get down to row 100, the hexadecimal notation takes over again, so that we see



The biggest number you can get is in the lower right hand corner, or FFFF in hexadecimal notation, or $(16 \times 16 \times 16 \times 16) = 65,536$

FFD	FFE	FFD
FFD	FFE	FFD
FFD	FFE	FFD

in normal numbers. This is commonly called 64000, or 64K in computerese. Each box can contain a piece of data called a BYTE, so the above describes a memory of 64K BYTES.

A sub-specification of memory is the type, which divides into two categories: that which is fixed; and that which can be changed.

FIXED - This kind of memory cannot be changed, added to, subtracted from, etc. It can only be looked at, or read. So they call it READ ONLY MEMORY, or ROM. If there is an Operating System included in a computer, it is in ROM. This allows the computer to be ready to operate as soon as you turn it on. In the Bally, there is an 8K ROM chip that contains the resident games, controller and keypad operations, etc. The addresses of the bytes in there are from 0000 to 1FFF in hex notation ($2 \times 16 \times 16 \times 16 = 8192$). As mentioned above, you can read the bytes that are permanently stored in the ROM addresses - here is line 0220 thru 022F:

[illegible]

exciting, isn't it? I have it on good authority that this translates to

X - S C O R E - # - O F - P L A	
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	1
19	1
20	1
21	1
22	1
23	1
24	1
25	1
26	1
27	1
28	1
29	1
30	1
31	1
32	1
33	1
34	1
35	1
36	1
37	1
38	1
39	1
40	1
41	1
42	1
43	1
44	1
45	1
46	1
47	1
48	1
49	1
50	1
51	1
52	1
53	1
54	1
55	1
56	1
57	1
58	1
59	1
60	1
61	1
62	1
63	1
64	1
65	1
66	1
67	1
68	1
69	1
70	1
71	1
72	1
73	1
74	1
75	1
76	1
77	1
78	1
79	1
80	1
81	1
82	1
83	1
84	1
85	1
86	1
87	1
88	1
89	1
90	1
91	1
92	1
93	1
94	1
95	1
96	1
97	1
98	1
99	1
100	1

Programs to read memory have been printed in Vol 1.

CHANGEBLES - In order to do useful work, there must be a memory area that can contain the program you want to operate, plus that data that you are going to utilize, input, or store. The computer provides a block of empty boxes in a space called RANDOM ACCESS MEMORY, or RAM. In some computers, the Operating System is stored on a tape or disc, and when you turn on the computer, you have to transfer the program from storage to a portion of the RAM that is set aside for this purpose (that is, there is no ROM).

In the Bally, when you insert the ^{Bally}~~Toy~~ BASIC cartridge, 4K of ROM is added which contains the Tiny BASIC Operating System, plus 1.5K of RAM for programming. In this instance, the amount of RAM is totally out of proportion to the amount of ROM, and that is what we are attempting to rectify with PROJECT ONE.

TO: Bally Arcade computer users.

An add-on circuit which improves the audio and video signals, optimizing for recording and/or transmission.

This add-on circuit gives the computer user a line level audio signal output and a composite video signal output. It is a lowest-possible-cost solution to a highest-possible-quality goal.

This add-on circuit was designed and prototyped by Dan Sandin; copied and documented by Phil Morton. For assistance contact Phil at (312) 666-5628, Chicago, Illinois.

For Bally Arcade computer users who are not connected into the ongoing Sandin IMAGE PROCESSOR cybernet, you probably should simply collect the parts (see PARTS LIST) and wire-wrap this circuit.

The following circuit diagram and 2X printed circuit board pictorial can be directly copied by Sandin IMAGE PROCESSOR builders using parts already on hand. The circuit is a slight variation on the "standard driver" used so frequently through-out the IP. We used 1/4 of a #217 board to build the circuit on.

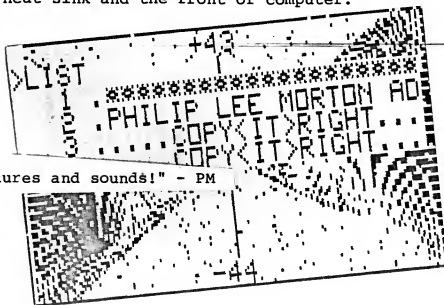
You can do a "neat" job by using either chassis-mount connectors, mounting them in the top plastic "fin", or cable-mount connectors by enlarging the RF Cable hole to run the audio and video cables out. We got away with using RG 174/U (coax) for both audio and video.

If you remove the RF Modulator from your computer then the BNC video out will deliver black-and-white composite video only; no color. This may be desirable for special applications which assume colorizing "down stream" in time.

Remove the five phillips-head screws on the bottom of the computer; the top plastic will now come off. Pull the RF Modulator off the 8 pin connector; solder to pins #1(video), #2(+10 volts), #3(audio). Pin #1 is closest to the heat sink and the front of computer.

Pin #8 = -5 volts
Pin #7 = B-Y
Pin #6 = R-Y
Pin #5 = +2.5 volts
Pin #4 = Chroma
Pin #3 = Audio
Pin #2 = +10 volts
Pin #1 = Video

"...enjoy your clean pictures and sounds!" - PM



PARTS LIST

Resistors:

1	47Ω	1/4 watt
2	10Ω	1/4 watt
2	100Ω	1/4 watt
2	10kΩ	1/4 watt

Capacitors:

```

1      .1uF 50wvdc cer. disc.
1      10uF 25wvdc electrolytic
1      100uF 25wvdc electrolytic
1      250uF 12wvdc electrolytic

```

Transistors:

```

1      2N4125  (PNP)
1      2N4123  (NPN)

```

Diodes:

2 1N914

Wire/cable:

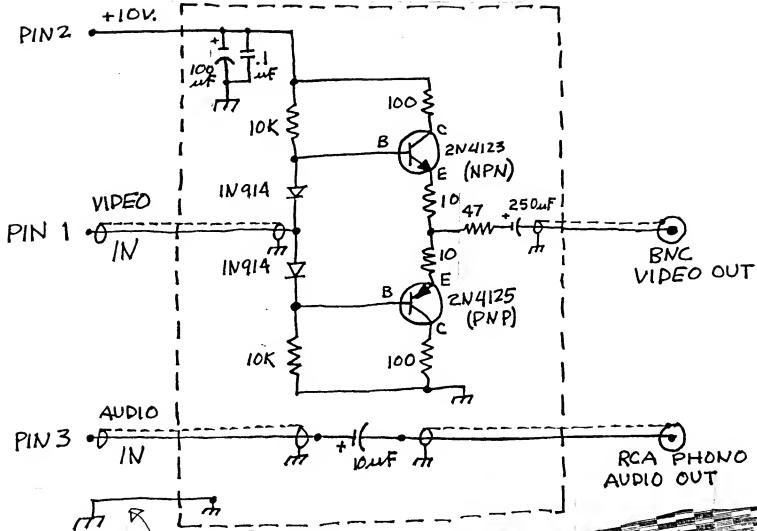
4 feet RG 174/U (coax)
2 feet hook-up, grounding, jumping wire

Connectors:

```
1 RCA phono-female (chassis or cable mount)
1 BNC video-female (chassis or cable mount)
```

PC Board:

1/4 of a #217 board (standard driver in IP)



SOLDER THIS GROUND TO THE BOTTOM OF GAME BOARD SHIELD.

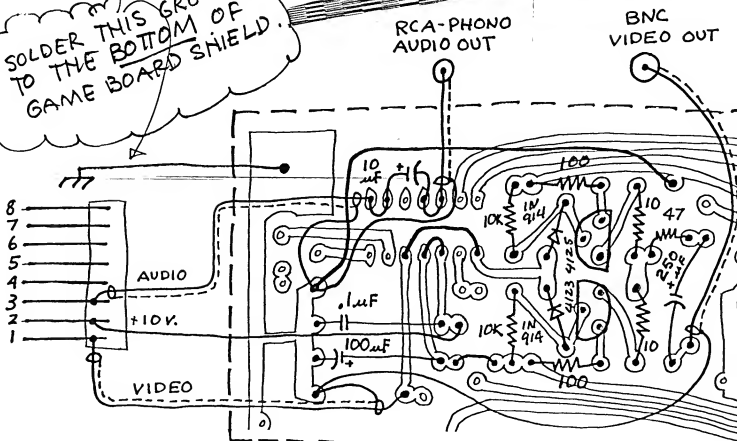




PHOTO: JANE VEEDER EYE

MUSIC BY VEEDER

RUN

AID
ELECTRONIC + VISUALIZATION

AID
ELECTRONIC + VISUALIZATION

AID
ELECTRONIC + VISUALIZATION

COMPUTER PROGRAMMING;
B.A.S.I.C. - ALLY INSIDE
TELEVISION."

"...ARTISTIC RESEARCH."
"...ARTISTIC RESEARCH."
"...ARTISTIC RESEARCH."
"...ARTISTIC RESEARCH."

"AN INTERACTIVE, HOME
COMPUTER-TELEVISION
RESEARCH SATELLITE..."

ELECTRONIC
VISUALIZATION
CENTER

This program is based on the "Space Invaders" game found in many arcades. It is a very challenging game, with difficulty levels from one to five. At level five, it is very easy to win, while level one is almost impossible. The major difference between this and the arcade game is speed. This one is considerably slower.

As for criticism, it would be better if the knob control (KN) were given as a better option. A modification is given in the program for its use, but it is not really satisfactory. Also, the spelling used is atrocious in the program and the documentation.

Also, perhaps a subroutine could be used to simulate explosions similar to the way the mother ship explodes. However, considering the memory limitations that the Bally has, much further improvement is difficult.

One final comment: Perhaps an option(re-programing) so that the landers would appear not just randomly but in sequence(like the real "Space Invaders") would make for a nicer display and add to the strategy aspect of the game.

Reviewed by Bill Rueger

BILL RUEGER
336 BEACH 30TH ST
FAR ROCKAWAY, NY 11691

Source Code GC Cassette Code 1 Program Code A BPRF4

BALLY PROGRAM REVIEW

Date: 12-7-79
Name of Program: Invasion Cassette Name: Same
Description: _____
Source: George Collins 30 Sierra Ave. Piedmont, CA 94611 Price: \$5.00
Reviewed by: Bill Rueger Age: 31

Circle score for each item 0 1 2 3 4 5 6 7 8 9 on scoring line.

PROGRAM DOCUMENTATION (PD)

No Instructions 0 1 2 3 4 5 6 7 8 9 Very Clear Description
Listings, Flowchart, Instructions

PROGRAM POLISH (PP)

Sloppy, Amateurish 0 1 2 3 4 5 6 7 8 9 Professional

USE OF SPECIAL FEATURES (USF)

Minimal Used 0 1 2 3 4 5 6 7 8 9 Great Use of Features
Graphics, Sound, Handles

LEVEL OF CHALLENGE (LC)

Not Challenging 0 1 2 3 4 5 6 7 8 9 Very Challenging

ORIGINALITY AND CREATIVITY (OC)

Adapted, Same Old Stuff 0 1 2 3 4 5 6 7 8 9 Totally Brilliant and Unique

LEVEL OF INTEREST (LI)

Boring 0 1 2 3 4 5 6 7 8 9 Fascinating

EDUCATIONAL VALUE (EV)

Little 0 1 2 3 4 5 6 7 8 9 Really Learn Facts
and Skills

EASE OF USE (EU)

Awkward, Inconvenient 0 1 2 3 4 5 6 7 8 9 Easy, Quick, Convenient

OVERALL VALUE (OV)

Almost Worthless 0 1 2 3 4 5 6 7 8 9 Everyone Should Buy

2-4
ARCADIAN
3626 MORRIE DR
SAN JOSE, CA 95127

I AM ANXIOUS FOR MORE NEWS ABOUT
THE FIDELITY TAKEOVER AND WHAT IT
MEANS IN TERMS OF A SWIFT INTRODUCTION
OF A KEYBOARD EXPANSION. KEEP UP THE
GOOD WORK!
MEANWHILE WILL YOU RUN THE
FOLLOWING ADS

SELL BALLY COMPUTER SYSTEM COMPLETE
WITH BALLY BASIC AND CASSETTE
INTERFACE, 4 PISTOL GRIPS
AND BLACKJACK, POKER, ACEY-DEUCEY
SEAWOLF AND MISSILE, STAR BATTLE
UNIT NEW IN NOV '79, COMPLETE \$285.00
CONTACT BILL FOWLER 42 BENT OAK TRAIL
FAIRPORT N.Y. 14450

SELL BRAND NEW HAND CONTROLS
10.00 EACH
4 FOR \$30.00

Thanks
Bill Fowler HOPEFUL

Please publish in latest issue

For Sale - like new Bally
Arcade. Includes 9 tapes
(including Bally Basic Cartridge),
Interface and 4 controllers.
Will take highest bid & notify
by telephone. Send bid and
phone # to

Steve Melton
Garden Grove Apts.
1601 - 9th Str. NW
Great Falls, Montana
59405

2-4

Robert Fabris
3626 Morrie Dr.
San Jose, Calif 95127

2-4
Ed
30 Sierra Avenue
Piedmont, CA, 94611
January 13, 1979

Robert Fabris
Staff, Boss, proof reader,
stapler, bolder, stamp
licker, tired
3626 Makrie Dr.
San Jose, CA 95127

Dear Mr. Fabris:

Please continue my subscription
to your excellent newsletter. Enclosed
is a check for ten dollars. Please
put this in your classified:
George Collins, 30 Sierra Ave.,
Piedmont, CA, 94611, is expanding
his line of software. Send a S.S.A.E.
for more information and a free
listing of the program U-Boat,

Sincerely Yours,
George Collins

u
January 21, 1980

call 1/24

Dear Robert:

Did you ~~XXX~~ hear that the Arcade was bought by Fidelity (sp?)???
...and the rumor is that "they" want to make it better; seems as though
all the Arcadians out here (where ever we are) would support that hope;
better-and-better! Towards this rumor, I would like to forward something
that is more fact than just rumor-fact; the enclosed documentation on
a very needed and useful add-on circuit for our Arcades. It works; we've
been using them for awhile; lot's of Arcade graphics have been broadcast
here in Chgo. on local PBS (Channel #11, WTTW) which have "proven" it so
to speak..."broadcast quality"!?

Did you see the latest issue of VIDEO magazine? They award the
Arcade for "best" ~~XXXX~~ audio/video effects!

Did you get the ~~XXXXXX~~ AXIOM videoprinter graphics I sent you awhile
back of Arcade stuff?

Somewhere in your editorial comments in near-future issue could you
please announce "congratulations" (or whatever) ~~the 12 new students~~
~~receiving the Digital Video course at the School of Art Institute~~
~~of Chicago. These students are the first to receive the~~
~~award for best student researchers~~

~~in Digital Video~~

CONGRADUATIONS!!! Or whatever, to the 12 newly created Arcadians in
the Digital Video course at the School of Art Institute of Chicago.

THEY ~~use~~ use six Arcades (w/BBasic) to learn programming. • - VISUAL ART STUDENTS

...oops, gotta' run. :RUN!

Yours:



Phil Morton; Assoc. Prof.
Video Area Research
School of Art Institute of Chicago

(312) 666-5628